# BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

## DEPARTMENT OF FOREIGN LANGUAGES

ÚSTAV JAZYKŮ

# SELECTED TYPES OF DC CIRCUIT ANALYSIS IN MATLAB PROGRAMMING LANGUAGE

VYBRANÉ DRUHY ANALÝZ STEJNOSMĚRNÝCH OBVODŮ V PROGRAMOVACÍM JAZYCE MATLAB

**BACHELOR'S THESIS**
BAKALÁŘSKÁ PRÁCE

**AUTHOR** **Hedvika Urbanová**
AUTOR PRÁCE

**SUPERVISOR** **Mgr. Ing. Eva Ellederová**
VEDOUCÍ PRÁCE

**BRNO 2017**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

# Bakalářská práce

bakalářský studijní obor **Angličtina v elektrotechnice a informatice**
Ústav jazyků

***Studentka:*** Hedvika Urbanová ***ID:*** 177787
***Ročník:*** 3 ***Akademický rok:*** 2016/17

**NÁZEV TÉMATU:**

**Vybrané druhy analýz stejnosměrných obvodů v programovacím jazyce MATLAB**

**POKYNY PRO VYPRACOVÁNÍ:**

Popište metody analýzy stejnosměrných obvodů a vysvětlete způsoby jejich výpočtu. Zdokumentujte postup programování těchto metod v programovacím jazyce MATLAB a zvažte možnost využití programů studenty.

**DOPORUČENÁ LITERATURA:**

1) Doňar, B., & Zaplatílek, K. (2004). MATLAB – tvorba uživatelských aplikací. Praha: BEN – technická literatura.

2) Wang, M. (2010). Understandable electric circuits. London: Institution of Engineering and Technology.

3) Myslík, J. (1979). Řešené příklady z analýzy elektrických obvodů. Plzeň: Vysoká škola strojní a elektrotechnická.

***Termín zadání:*** 6.2.2017 ***Termín odevzdání:*** 2.6.2017

***Vedoucí práce:*** Mgr. Ing. Eva Ellederová
***Konzultant:*** doc. Ing. Jana Kolářová, Ph.D.

**doc. PhDr. Milena Krhutová, Ph.D.**
*předseda oborové rady*

## Abstract

This bachelor thesis deals with methods of a DC circuit analysis. Both universal and special methods are discussed. All methods are theoretically determined. Calculation procedure is explained and advantages and disadvantages of each method are considered. Universal methods are explained by means of a problem identical for each method. Next, MATLAB is introduced as a programming language. Four of the discussed methods are written into the programming language MATLAB 2014a, namely the loop current method, the node voltage method, Kirchhoff's laws, and the superposition method. The loop current method as well as Kirchhoff's laws are programmed for two to four loops. The node voltage method is programmed for two to four nodes. The superposition method is programmed only for two loops. A graphical user interface is defined and designed for the program of Kirchhoff's laws. Finally, the use of the programs by students is considered.



## Key words

DC circuit analysis, programming language, MATLAB, loop current method, node voltage method, Kirchhoff's laws, simplification method, graphical user interface



## Abstrakt

Tato bakalářská práce se zabývá metodami analýz elektrických obvodů a to jak metodami univerzálními, tak speciálními. Metody byly v práci teoreticky definovány. Následně byl vysvětlen postup při počítání a zhodnoceny jejich výhody a nevýhody. Univerzální metody byly vysvětleny na příkladu, který byl pro všechny metody identický. MATLAB byl představen jako programovací jazyk. Čtyři z metod byly přepsány do programovacího jazyka MATLAB 2014a. Naprogramovány byly metody smyčkových proudů, uzlových napětí, Kirchhoffovy zákony a metoda superpozice. Metoda smyčkových proudů byla stejně jako Kirchhoffovy zákony naprogramována pro dvě až čtyři smyčky. Metoda uzlových napětí byla pak naprogramována pro dva až čtyři uzly. Metoda superpozice byla naprogramována pouze pro dvě smyčky. Grafické uživatelské rozhraní bylo definováno a vytvořeno pro program Kirchhoffových zákonů. V závěru projektu bylo zváženo využití programů studenty.



## Klíčová slova

Analýza stejnosměrných elektrických obvodů, programovací jazyk, MATLAB, metoda smyčkových proudů, metoda uzlových napětí, Kirchhoffovy zákony, metoda postupného zjednodušování, grafické uživatelské rozhraní

# PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma *Vybrané druhy analýz stejnosměrných elektrických obvodů v programovacím jazyce MATLAB* jsem vypracovala samostatně pod vedením vedoucí bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .............................                                   ...................................

(podpis autora)

## PODĚKOVÁNÍ

Ráda bych poděkovala vedoucí mé bakalářské práce Mgr. Ing. Evě Ellederové a doc. Ing. Janě Kolářové, Ph.D. za jejich cenné rady a čas.

URBANOVÁ, H. *Vybrané druhy analýz stejnosměrných obvodů v programovacím jazyce MATLAB.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 49 s. Vedoucí bakalářské práce Mgr. Ing. Eva Ellederová.

# Table of Contents

# 1 Introduction

This Bachelor's thesis will deal with all types of DC circuit analysis and their transfer into a programming language. DC analysis of a complicated circuit may be very difficult to calculate by hand, in particular the loop current method and the voltage node method. In a circuit with more than four loops, the determinant of the matrix of the fourth order has to be calculated when the loop current method is used. In complicated circuits, many errors in computation can be made. There is a very useful solution for students to verify their results. Most students of the Faculty of Electrical Engineering and Communication can work with MATLAB. I decided to deal with this topic because I consider the methods of DC circuit analysis as very useful verification of the results, for example before an exam.

The theoretical part of this Bachelor's thesis will provide a comprehensive overview of all methods used for DC circuit analysis and their history. It will focus on calculations and their purpose. One particular problem identical for every universal method will be used for an explanation of calculations. The theoretical part will also briefly describe the use of the methods in electrical engineering practice. MATLAB will then calculate the same problem in the practical part of this thesis so that we could decide if it is calculated accurately.

MATLAB will be introduced as a programming language in the practical part of this bachelor's thesis. Some of the basic MATLAB functions and processes of writing programs will be discussed. These programs will use the methods to calculate all values of current and voltage in the particular circuit. Each part of the program will be explained step by step. The practical part will focus on writing down the loop current method and the node voltage method. It will also provide an example of the outputs of the program.

Finally, calculations by hand and calculations using the MATLAB program will be analysed and compared in the conclusion of this Bachelor's thesis. MATLAB version 2014a will be used for programming. Older versions may have some problems, for example with the program in the graphical user interface.

# 2 Methods of DC Circuits Analysis

All analyses of DC circuits are based on the three important laws. These are: Kirchhoff's current law, Kirchhoff's voltage law, and Ohm's law. The definition of Kirchhoff's first law can be written in this equation: $\sum I = 0$ (in words: Algebraic sum of all currents in the node is equal to zero). The Kirchhoff's second law is similar and can be defined by this simple equation: $\sum U = 0$ (in words: Algebraic sum of all voltages in the closed independent loop is equal to zero). At last, Ohm's law is defined by the usual mathematical equation describing the relationship between current, voltage, and resistance: $U = R \cdot I$.

We should also mention all types of methods of DC analysis and the main reason why they are helpful. These methods are frequently used to identify every voltage and current in DC circuit. The reason why we need to calculate these values is very clear. In houses with electric appliances, it is necessary to calculate all values of resistance, current, and voltage to avoid a blackout caused by short circuit. Every electrician has to decide on the base of these values which circuit breaker has to be used.

The methods of DC analysis can be divided into two different groups: universal and special. They will be discussed in the following chapters.

## 2.1 History of DC circuit analysis

Electricity is one of quite new branches in physics. Its development took place after the discovery of the first usable source of the constant electric current – voltaic pile – in 1800. The foundations of electrostatics were discovered even sooner. Charles Coulomb discovered the force between two electric charges in 1785. In the 1820s André Marie Ampére laid the foundations of electromagnetism and electrodynamics. The German scientist Georg Simon Ohm formulated the law of dependence between electric current, voltage and resistance in 1826. The English scientist Michael Faraday discovered electromagnetic inductance in 1831.

**Ohm's law:** Although, the primacy in the formulation of Ohm's law belongs to Henry Cavendish, it was independently rediscovered and first published by Georg Simon Ohm.

The most prominent Georg Simon Ohm's work in the history of electrical engineering is his article "Über Leitungsfähigkeit der Metalle für Elektrizität" (1825). Ohm showed that the ratios in a simple electrical circuit are uniquely determined by three variables: the current, the voltage of the source and the resistance, which is directly proportional to the length of the conductor and inversely proportional to its cross-section. Even though Ohm's first attempt was incorrect, he continued to work on the problem. It turned out that the main cause was that the voltaic pile was used as a source of voltage and Ohm did not count on its voltage dropping during the measurements. According to Farrell (1976, p. 704):

> At first Ohm used a voltaic pile as his source, but it gave him a lot of trouble. Not only was there a current surge every time he connected it but the generation of gas in the pile reduced the current as the experiment proceeded. With much trouble Ohm managed to get consistent results which he published in 1825 and it was his editor who gave him a new crucial idea. Poggendorf suggested he use Seebeck's recently discovered thermoelectric effect to generate the voltaic force as this would not change with time nor give current surges. Ohm started out again in 1826, the crucial year, to re-experiment using a thermocouple – this time with success.

**Kirchhoff's laws:** Under the leadership of prof. Franz Neumann, Gustav Robert Kirchhoff wrote a paper in which he formulated laws that were later associated with his name. Kirchhoff's paper received a well-deserved award. He modified his paper, which later became the base of his dissertation thesis. This thesis was very popular with various physicists. Kirchhoff was very devoted to theoretical physics. He experimented a lot. Kirchhoff laid the foundations of the current theory of electrical circuits in a relatively narrow period of time 1845-1849. In the next period of his life he did not deal with electrical and magnetic phenomena. He was dealing with optics and the theory of radiation. He was motivated by Joseph von Fraunhofer's discovery of spectral lines.

**Thévenin's theorem:** It was a principle discovered by Hermann von Helmholtz in 1853. In 1883 it was rediscovered by a French telegraph engineer Léon Charles Thévenin.

**Norton's theorem:** It was discovered by Edward Lawry Norton in the 1920s. According to Maloberti (2016, pp. 38-39):

Although his primary interests were in network theory, acoustical systems, electromagnetic apparatus, and data transmission a communications circuit theory and the transmission of data at high speeds over telephone lines, Edward L. Norton is universally recognised for development of the dual of Thévenin's equivalent circuit. Norton in the early 1920s was one of the first scientists who applied the Thévenin's equivalent circuit to simplify networks; then realised that in some cases it can be convenient to use an alternative method.

**History of DC Circuit Analysis in MATLAB**

In 1999 John Okyere Attia published his book "Electronics and Circuit Analysis using MATLAB". In this book he introduces MATLAB and its possibilities and defines the procedures of creating programs in this programming environment and programming language. The DC circuit analysis is though only one short chapter in this book. He created program for node voltage method and loop current method.

In 2014 a group of scientists including Dr. Yousuf Asad and Mr. William Lehman created a short textbook for students with procedures, how to do circuit analysis in MATLAB and Simulink.

## 2.2 Universal Methods

Universal methods are based on matrix algebra. These methods are universal for every DC circuit with linear components such as resistors, current sources, and voltage sources. Every method will be explained using a calculation of a problem. A circuit which will be calculated is in the picture below (Figure 1.). In these methods Ohm's law and Kirchhoff's laws will be used.

The values of resistances and voltages are known as indicated:

$R_1 = 10\ \Omega$

$R_2 = 20\ \Omega$

$R_3 = 25\ \Omega$

$R_4 = 30\ \Omega$

$R_5 = 15\ \Omega$

$V_1 = 2\ V$

$V_2 = 5\ V$

$V_3 = 3\ V$

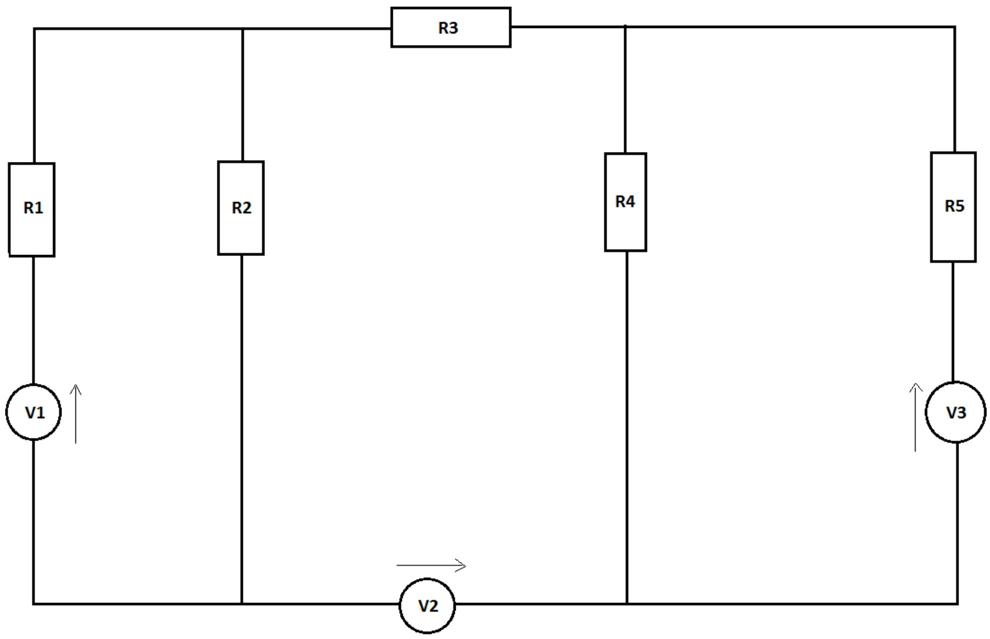


*Figure 1.* Calculated circuit.

### 2.2.1 Kirchhoff's Laws

A system of specific numbers of equations (the number is equal to the number of loops in the circuit) or an augmented matrix with the same number of rows is used in this method. We get the result (all currents in every branch and voltage drop in every resistor) by editing the matrix into a triangle form.

**Kirchhoff's current law**

According to Herrick (2003, p. 176), "Kirchhoff's current law states that the sum of currents into node has to be equal to current coming out of that node".

This principle can be stated as: $\sum I_{\text{into node}} = \sum I_{\text{out of node}}$

**Kirchhoff's voltage law**

According to Herrick (2003, p. 205), "Kirchhoff's voltage law essentially states that as you traverse any closed loop of a circuit (adding voltage rises and subtracting voltage falls), the net result is 0 V difference".

**Problem**

First, directions of the currents in the two main nodes have to be determined (see fig. 2).

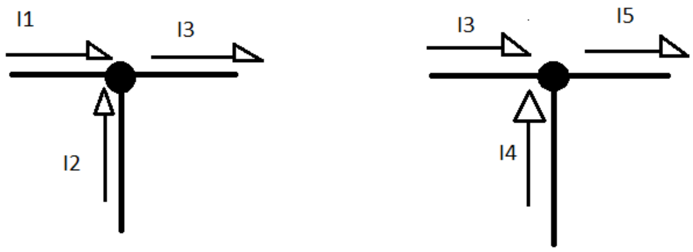


*Figure 2.* Determined directions.

Then the two equations using Kirchhoff's current law have to be written in this case (see fig. 1). Next Kirchhoff's voltage law for all three loops is used. After this step five equations with five variables are obtained, which implies that a set of five equations has to be calculated. After making make the long calculation on a piece of paper (it can even take up one A4 page), we get the following result:

$$I_1 + I_2 - I_3 = 0 \quad (1)$$

$$I_3 + I_4 - I_5 = 0 \quad (2)$$

$$R_1 \bullet I_1 - R_2 \bullet I_2 = V_1 \quad (3)$$

$$R_2 \bullet I_2 - R_3 \bullet I_3 - R_4 \bullet I_4 = - V_2 \quad (4)$$

$$\underline{R_4 \bullet I_4 - R_5 \bullet I_5 = - V_3} \quad (5)$$

Where:

$I_1$, $I_2$, $I_3$, $I_4$, $I_5$ are the unknown currents for loops 1 through 3;

$R_1$, $R_2$, $R_3$, $R_4$, $R_5$ are known resistances;

$V_1$, $V_2$, $V_3$ are known voltages of sources 1 through 3.

By substitution of given values, we get:

$I_1 + I_2 - I_3 = 0$

$I_3 + I_4 - I_5 = 0$

$10 \cdot I_1 - 20 \cdot I_2 = 2$

$20 \cdot I_2 - 25 \cdot I_3 - 30 \cdot I_4 = -5$

$\underline{30 \cdot I_4 - 15 \cdot I_5 = -3}$

$I_1 = (2+20 \cdot I_2)/10$

$\underline{I_5 = (-1-10 \cdot I_4)/5}$

$30 \cdot I_2 - 10 \cdot I_3 = -2$

$\underline{20 \cdot I_2 - 25 \cdot I_3 + 30 \cdot I_4 = -5}$

$95 \cdot I_3 - 90 \cdot I_4 = -11$

$\underline{5 \cdot I_3 + 15 \cdot I_4 = -1}$

$\underline{\underline{I_1 = -0.024 \text{ A} = -24 \text{ mA}}}$

$\underline{\underline{I_2 = 0.112 \text{ A} = 112 \text{ mA}}}$

$\underline{\underline{I_3 = -0.136 \text{ A} = -136 \text{ mA}}}$

$\underline{\underline{I_4 = 0.021 \text{ A} = 21 \text{ mA}}}$

$\underline{\underline{I_5 = -0.157 \text{ A} = -157 \text{ mA}}}$

Voltage losses for each resistor is then computed as follows:

$\underline{\underline{V_{R1} = R_1 \cdot I_1 = -0.24 \text{ V}}}$

$\underline{\underline{V_{R2} = R_2 \cdot I_2 = 2.24 \text{ V}}}$

$\underline{\underline{V_{R3} = R_3 \cdot I_3 = -3.4 \text{ V}}}$

$\underline{\underline{V_{R4} = R_4 \cdot I_4 = 0.64}}$

$\underline{\underline{V_{R5} = R_5 \cdot I_5 = -2.36 \text{ V}}}$

### 2.2.2 Loop Current Method

For the explanation of the loop current method, Chen’s (2004, p. 7) description is used:

> In this method, one distinct current variable is assigned to each independent loop. The element currents are then calculated in the terms of the loop currents. Using the element currents and values, element voltages are calculated. After these calculations, Kirchhoff’s voltage law is applied to each of the loops, and the

resulting equations are solved for loop currents. Using the loop currents, element currents and voltages are then determined.

Equations of the matrices are expressed and the value of the determinant of the main matrix is computed. The next step is the calculation of the determinants of the other matrices by means of swapping the columns of the original matrix and the voltage matrix.

**Problem**

First, we have to determine the direction of the loop currents (see fig. 3).

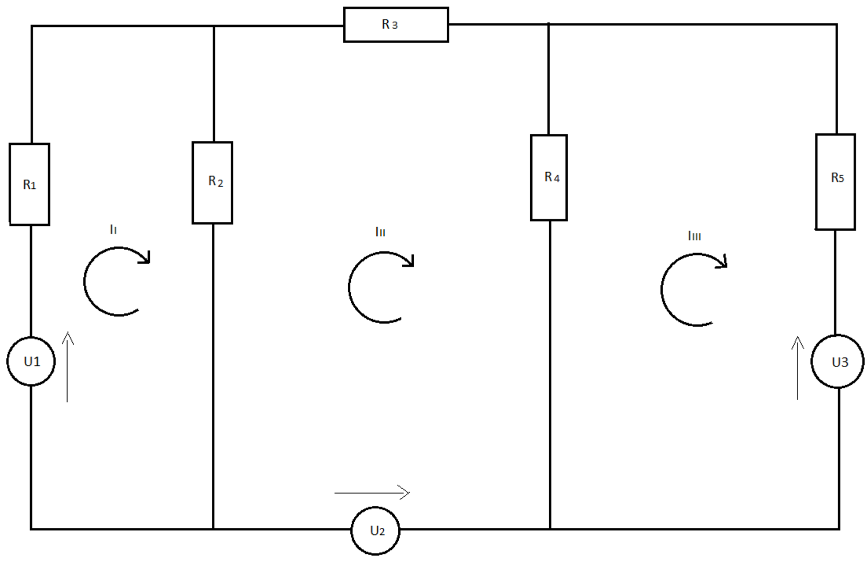


*Figure 3*. Circuit with loop currents.

Equations (eq. 6, 7, 8, 9) can be expressed in matrix form as:

$$\begin{bmatrix} R1+R2 & -R2 & 0 \\ -R2 & R2+R3+R4 & -R4 \\ 0 & -R4 & R4+R5 \end{bmatrix} \cdot \begin{bmatrix} I_I \\ I_{II} \\ I_{III} \end{bmatrix} = \begin{bmatrix} V_1 \\ -V_2 \\ -V_3 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} V_1 & -R2 & 0 \\ -V_2 & R2+R3+R4 & -R4 \\ -V_3 & -R4 & R4+R5 \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} R1+R2 & V_1 & 0 \\ -R2 & -V_2 & -R4 \\ 0 & -V_3 & R4+R5 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} R1+R2 & -R2 & V_1 \\ -R2 & R2+R3+R4 & -V_2 \\ 0 & -R4 & -V_3 \end{bmatrix} \tag{9}$$

Now the determinants of the original matrix and the three other matrices have to be enumerated:

$$det\left(\begin{bmatrix} 30 & -20 & 0 \\ -20 & 75 & -30 \\ 0 & -30 & 45 \end{bmatrix}\right) = 56250$$

$$det\left(\begin{bmatrix} 2 & -20 & 0 \\ -5 & 75 & -30 \\ -3 & -30 & 45 \end{bmatrix}\right) = -1350$$

$$\det = \left(\begin{bmatrix} 30 & 2 & 0 \\ -20 & -5 & -30 \\ 0 & -3 & 45 \end{bmatrix}\right) = -7650$$

$$\det = \left(\begin{bmatrix} 30 & -20 & 2 \\ -20 & 75 & -5 \\ 0 & -30 & -3 \end{bmatrix}\right) = -8850$$

The loop currents can be calculated by means of the division of the determinants.

$I_{L1}$ = -1350/56250 = - 0.024 = - 24 mA

$I_{L2}$ = -7650/56250 = - 0.136 A = - 136 mA

$I_{L3}$ = -8850/56250 = - 0.157 A = -157 mA

Then the values of the currents are calculated with respect to the original circuit:

$I_1 = I_{L1}$ = -24 mA

$I_2 = I_{L1} – I_{L2}$ = -24+136 = 112 mA

$I_3 = I_{L2}$ = -136 mA

$I_4 = I_{L2} – I_{L3}$ = -136+157 = 21 mA

$I_5 = I_{L3}$ = -157 mA

To complete the analysis, voltage loss for each resistor has to be calculated.

$V_{R1} = R_1 \cdot I_1$ = -0.24 V

$V_{R2} = R_2 \cdot I_2 = 2.24$ V

$V_{R3} = R_3 \cdot I_3 = -3.4$ V

$V_{R4} = R_4 \cdot I_4 = 0.64$

$V_{R5} = R_5 \cdot I_5 = -2.36$ V

### 2.2.3 Node Voltage Method

The explanation of the node voltage method calculation process is given by Chen (2004, p. 108):

> The node voltage method uses a new type of variable called the node voltage. The "node voltage equations" or, more simply, the "node equations", are set of simultaneous equations that represent given electric circuit. The unknown variables of the node voltage equations are the node voltages. After solving the node voltage equations, we determine the values of the node voltages.

Equations can be written in matrices in the same way as in the loop current method. The main difference is that conductivities instead of resistances are used for calculations. But before the calculation starts, the circuit has to be recalculated (see fig. 4).

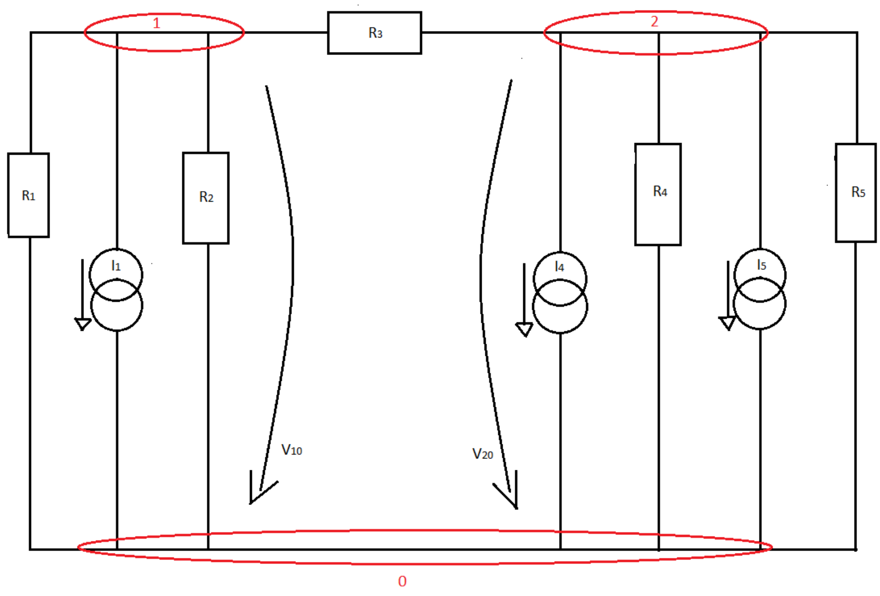


*Figure 4*. Recalculated circuit for the node voltage method.

**Problem**

Given values are still the same. Node voltages are in the circuit (see fig. 4) named V10 and $V_{20}$. Matrix equation (eq. 10) for the node voltage method is given as:

$$\begin{bmatrix} \frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3} & \frac{-1}{R3} \\ \frac{-1}{R3} & \frac{1}{R3}+\frac{1}{R4}+\frac{1}{R5} \end{bmatrix} \cdot \begin{bmatrix} V10 \\ V20 \end{bmatrix} = \begin{bmatrix} -I1 \\ -I2-I3 \end{bmatrix} \tag{10}$$

From the following matrices (eq. 11, 12, 13) determinants have to be made:

$$\begin{bmatrix} \frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3} & \frac{-1}{R3} \\ \frac{-1}{R3} & \frac{1}{R3}+\frac{1}{R4}+\frac{1}{R5} \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} -I1 & \frac{-1}{R3} \\ -I2-I3 & \frac{1}{R3}+\frac{1}{R4}+\frac{1}{R5} \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} \frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3} & -I1 \\ \frac{-1}{R3} & -I2-I3 \end{bmatrix} \tag{13}$$

The following determinants are then used to compute values of node voltages $V_{10}$ and $V_{20}$:

$$det \begin{bmatrix} \frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3} & \frac{-1}{R3} \\ \frac{-1}{R3} & \frac{1}{R3}+\frac{1}{R4}+\frac{1}{R5} \end{bmatrix} = 0.0250$$

$$det \begin{bmatrix} -I1 & \frac{-1}{R3} \\ -I2-I3 & \frac{1}{R3}+\frac{1}{R4}+\frac{1}{R5} \end{bmatrix} = -0.0560$$

$$det \begin{bmatrix} \frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3} & -I1 \\ \frac{-1}{R3} & -I2-I3 \end{bmatrix} = -0.1410$$

$V_{10}$ = -2.24 V

$V_{20}$ = -5.64 V

Final voltages on resistors are calculated from the original circuit (see fig. 1):

$V_{R1} = V_{10}+V_1$ = -0.24 V

$V_{R2} = V_{10}$ = -2.24 V

$V_{R3} = V_{20}-V_{10}$ = -3.4 V

$V_{R4} = V_{20}+V_2$ = -0.64

$V_{R5} = V_{20}+V_3$ = 2.36 V

Final currents can be calculated as follows:

$I_1 = V_{R1}/R_1$ = -0.24/10 = -24 mA

$I_2 = V_{R1}/R_2$ = -2.24/20 = -112 mA

$I_3 = V_{R3}/R_3$ = -3.4/25 = -136 mA

$I_4 = V_{R4}/R_4$ = -0.64/30 = -21 mA

$I_5 = V_{R5}/R_5$ = 2.36/15 = 157 mA

The calculated values of the currents are the same as the values calculated by the loop current method.

### 2.2.4 Superposition Method

The superposition method is based on the superposition theorem, which states that the algebraic sum of the individual solutions is equal to the general solution. A disadvantage of this method is that it can be calculated only with linear components.

When a circuit with more than one source is analysed, it is calculated with only one source and the other sources are added to a short circuit. It is calculated for every source in the same way and then the sum of all solutions is computed.

**Problem**

For the last method the same circuit (see fig. 1) and values of resistances and sources are used.

First, we start with the circuit with voltage source number 1. The circuit is shown in the following picture (see fig. 5).

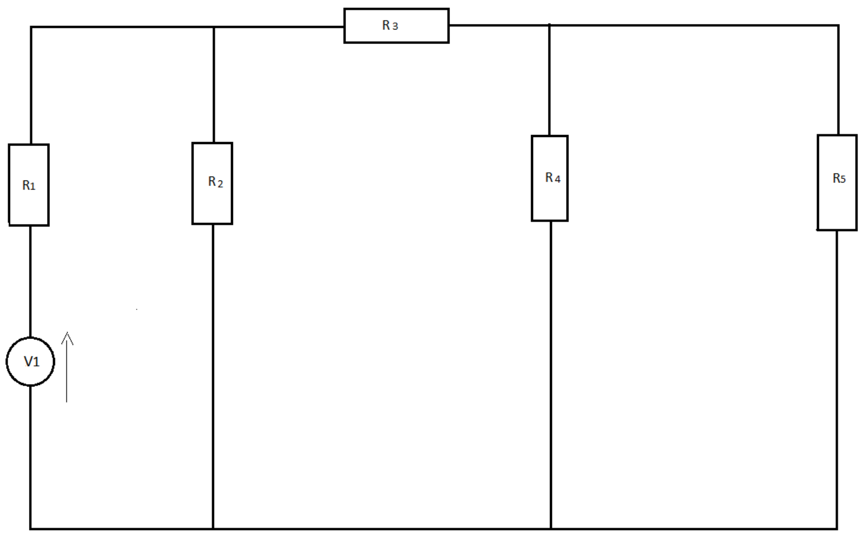


*Figure 5.* Circuit for the first source.

First, the simplification method has to be used and the sum of all resistances has to be computed:

$R_I' = 1/R_4 + 1/R_5 = 10\ \Omega$

$R_{II}' = R_I' + R_3 = 35\ \Omega$

$R_{III}' = 1/R_{II}'+1/R_2 = 140/11\ \Omega$

$R_{IV}' = R_{III}' + R_1 = 250/11\ \Omega$

Now currents and voltage losses for each resistor are calculated:

$I_1' = V/R_{IV} = 0.0880$ A

$I_2' = V_{R2}'/R_2 = 0.0560$ A

$I_3' = I_1' - I_2' = 0.0320$ A

$I_4' = V_{R4}'/R_4 = 0.0107$ A

$I_5' = V_{R5}'/R_5 = 0.0213$ A

$V_{R1}' = R_1 \bullet I_1' = 0.88$ V

$V_{R2'} = V - V_{R1}' = 1.12$ V

$V_{R3}' = I_3' \bullet R_3 = 0.80$ V

$V_{R4}' = V_{R5}' = V_{R2}' - V_{R3}' = 0.32$ V

The next step is calculation of the circuit for the second source (see fig. 6).

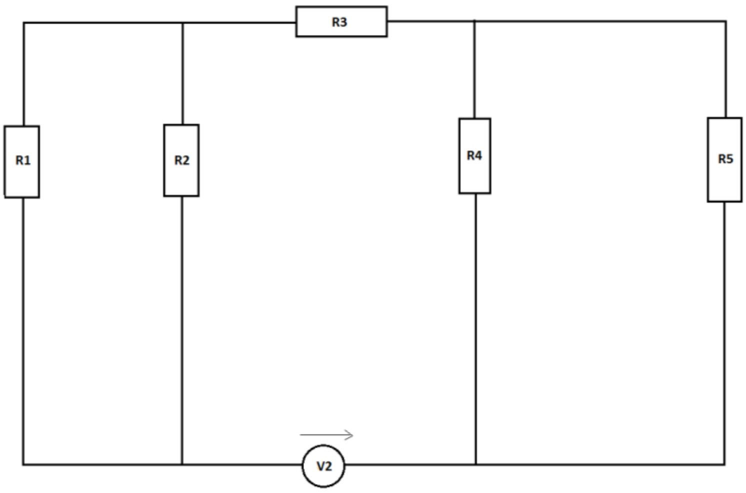


*Figure 6.* Circuit for the second source.

First, the sum of all resistances has to be calculated:

$R_I'' = 1/R_1 + 1/R_2 = 20/3\Omega$

$R_{II}'' = 1/R_4 + 1/R_5 = 10\Omega$

$R_{III}'' = R_I'' + R_3 + R_{II}'' = 125/3\Omega$

$I_3'' = V/R_{III}'' = 3/25$ A

$V_{R3}'' = R_3 \bullet I_3'' = 3$ V

$V_I'' = V_{R1}'' = V_{R2}'' = I_3'' \bullet R_I'' = 4/5$ V

$V_{II}'' = V_{R4}'' = V_{R5}'' = I_3'' \bullet R_{II}'' = 6/5$ V

$I_1'' = V_I''/R_1 = 2/25$ A

$I_2'' = V_I''/R_2 = 1/25$ A

$I_4'' = V_{II}''/R_4 = 1/25$ A

$I_5'' = V_{II}''/R_5 = 2/25$ A

Circuit with the second source (see fig. 7) has to be analysed in the next step.

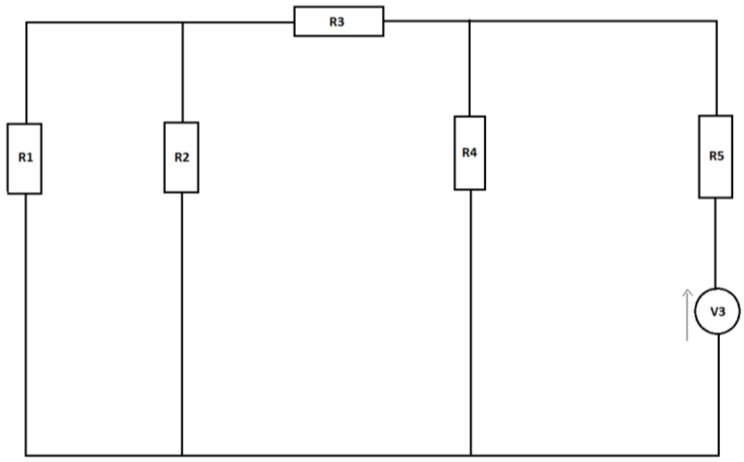


*Figure 7.* Circuit for the third source.

$R_I''' = 1/R_1 + 1/R_2 = 20/3\ \Omega$

$R_{II}''' = R_I''' + R_3 = 95/3\ \Omega$

$R_{III}''' = 1/R_{II}'''+1/R_4 = 2850/185\ \Omega$

$R_{IV}''' = R_{III}''' + R5 = (2850/185+15)\ \Omega$

$I_5''' = V/R_{IV}''' = 0.0987$ A

$I_4''' = V_{R4}'''/R_4 = 0.0507$ A

$I_3''' = I_5'''- I_4''' = 0.048$ A

$I_2''' = V_{R2}'''/R_2 = 0.032$ A

$I_1''' = V_{R1}'''/R_1 = 0.016$ A

$V_{R5}''' = I_5'''\bullet R_5 = 1.48$ V

$V_{R4}''' = V-V_{R5}''' = 1.52$ V

$V_{R3}''' = I_3'''\bullet R_3 = 1.2$ V

$V_{R2}'''=V_{R1}'''= V_{R4}'''-V_{R3}''' = 0.32$ V

Now the sum of all currents and the sum of all voltages have to be calculated. Furthermore, the directions of currents and voltages have to be incorporated.

$\underline{\underline{I_1 = -I_1' + I_1'' + I_1''' = -0.088+1/25+0.032 = 24 \text{ mA}}}$

$\underline{\underline{I_2 = I_2' + I_2'' + I_2''' = 0.056+1/25+0.016 = 112 \text{ mA}}}$

$\underline{\underline{I_3 = -I_3' + I_3'' + I_3''' = -0.032+3/25+0.048 = 136 \text{ mA}}}$

$I_4 = I_4' - I_4'' + I_4''' = 0.0107-1/25+0.0507 = 21$ mA

$I_5 = -I_5' + I_5'' + I_5''' = -0.0213+2/25+0.0987 = 157$ mA

$V_1 = -V_1' + V_1'' + V_1''' = -0.88+4/5+0.32 = 0.24$ V

$V_2 = V_2' + V_2'' + V_2''' = 1.12 + 4/5 + 0.32 = 2.24$ V

$V_3 = V_3' + V_3'' + V_3''' = -0.8 + 3 + 1.2 = 3.4$ V

$V_4 = V_4' + V_4'' + V_4''' = 0.32 - 6/5 + 1.52 = 0.64$ V

$V_5 = V_5' + V_5'' + V_5''' = -0.32 + 6/5 + 1.48 = 2.36$ V

## 2.3 Special Methods

Special methods are not suitable for every circuit or every component. They are usually combined with one of the universal methods to simplify calculation.

**Simplification Method**

This method is used to simplify the circuit. All linear components such as resistors, inductors, or capacitors are added up together. A disadvantage of this method is that it can be used only in a circuit with one source. In this method, we use two formulas for the parallel or series configuration. Following formulas are used for resistances.

Parallel: $1/R = 1/R_1+1/R_2$

Series: $R = R_1+R_2$

**Transformation**

This method is used to simplify the circuit. A transformation changes one part of the circuit in order to simplify its solution. There are two types of transformation – a delta network and a star network (see fig. 8).

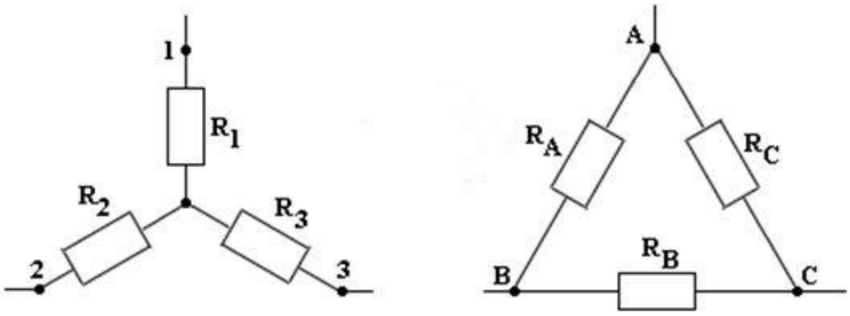


*Figure 8.* Types of transformation: A star network (left) and a delta network (right).

Formulas of a transformation from a delta to star network are following:

$$R_1 = \frac{R_A R_C}{R_A + R_B + R_C} \tag{14}$$

$$R_2 = \frac{R_A R_B}{R_A + R_B + R_C} \tag{15}$$

$$R_3 = \frac{R_B R_C}{R_A + R_B + R_C} \tag{16}$$

Formulas of a transformation from a star to delta network are following:

$$R_A = R_1 + R_2 + \frac{R_1 R_2}{R_3} \tag{17}$$

$$R_B = R_2 + R_3 + \frac{R_2 R_3}{R_1} \tag{18}$$

$$R_C = R_1 + R_3 + \frac{R_1 R_3}{R_2} \tag{19}$$

**Thévenin's and Norton's Theorems**

The Thévenin's theorem about a substituted source holds that any complicated linear circuit can be replaced by a circuit with one real source of voltage, which is connected to two braces. This method can be used only in circuits where only one branch needs to be calculated. The Norton's theorem for electric circuits holds that any system of linear source of voltage, sources of current and resistors with two braces is electrically equivalent to the ideal source of current with an ideal resistor in parallel connection.

**Method of Proportional Quantities**

This method is suitable especially for simple linear circuits with only one independent source. It is based on proportionality of quantities, where resistances are constants of proportionality.  First voltage or current is determined in one part of the circuit. Then all voltages and currents in the circuit are calculated. In the last step, values are recalculated with respect to the value of the original source.

# 3 MATLAB Programming Language

MATLAB is an abbreviation of “Matrix Laboratory”. Zaplatílek (2014) describes MATLAB as extended software used for technical calculations, plotting graphs, processing signals, programming of applications, measurement, etc. Zaplatílek (2004) says that we can work with this system in many different ways. Before the start of programming, a circuit has to be chosen. According to MathWorks (n.d.):

> The MATLAB platform is optimized for solving engineering and scientific problems. The matrix-based MATLAB language is the world’s most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. A vast library of prebuilt toolboxes lets you get started right away with algorithms essential to your domain. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together.

Circuits for the loop circuit method and the node voltage method are most commonly used. Everyone who wants to use this program can decide which resistances or sources will be used in the circuit they will need to calculate.

## 3.1 Loop Current Method in MATLAB

First, matrices have to be defined. The following function defines the matrix of resistances:

```
function [R] = RmatrixLCM(R1,R2,R3,R4,R5,R6)
R = [R2+R1,-R2,0;-R2,R2+R4+R3+R6,-R4;0,-R4,R4+R5];
end
```

The following function defines the matrix of all voltages:

```
function [V] = VmatrixLCM(V1,V2,V3,V4,V5,V6)
V = [V1-V2;V2+V3-V4-V6;V3-V5];
end
```

The function *determinant0* computes the determinant of the matrix of resistances:

```
function D = determinant0(R1,R2,R3,R4,R5,R6)
Ru0 = RmatrixLCM(R1,R2,R3,R4,R5,R6);
D = det(Ru0);
end
```

The function *determinant1* changes the first column of the resistance matrix with the voltage matrix. The determinant is computed from the result of the function:

```
function D1 = determinant1(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6)
Ru1 = RmatrixLCM(R1,R2,R3,R4,R5,R6);
V = VmatrixLCM(V1,V2,V3,V4,V5,V6);
for i = 1:3
    Ru1(i,1) = V(i,1);
end
D1 = det(Ru1);
end
```

The function *determinant2* changes the second column of the resistance matrix with the voltage matrix. The determinant is computed from the result of the function.

```
function D2 = determinant2(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6)
Ru2 = RmatrixLCM(R1,R2,R3,R4,R5,R6);
V = VmatrixLCM(V1,V2,V3,V4,V5,V6);
for i = 1:3
    Ru2(i,2) = V(i,1);
end
D2 = det(Ru2);
End
```

The function *determinant3* changes the second column of the resistance matrix with the voltage matrix. The determinant is computed from the result of the function.

```
function D3 = determinant3(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6)
Ru3 = RmatrixLCM(R1,R2,R3,R4,R5,R6);
V = VmatrixLCM(V1,V2,V3,V4,V5,V6);
for i = 1:3
    Ru3(i,3) = V(i,1);
end
D3 = det(Ru3);
end
```

The function *LcurrentI* calls the functions of *determinants 0* and *1* to compute their ratio.

```
function Is1 = LcurrentI(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6)
D = determinant0(R1,R2,R3,R4,R5,R6);
D1 = determinant1(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6);
Is1 = D1/D;
end
```

The function *LcurrentII* calls the functions of *determinants 0* and *2* to compute their ratio.

```
function Is2 = LcurrentII(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6)
D = determinant0(R1,R2,R3,R4,R5,R6);
D2 = determinant2(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6);
Is2 = D2/D;
end
```

The function *LcurrentII* calls the functions of *determinants 0* and *3* to compute their ratio.

```
function Is3 = LcurrentIII(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6)
D = determinant0(R1,R2,R3,R4,R5,R6);
D3 = determinant3(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6);
Is3 = D3/D;
end
```

The function *finalcurrents* calls the functions of the loop currents and calculates from them the currents in each branch.

```
function [I1,I2,I3,I4,I5,I6] =
finalcurrents(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6)
Is1 = LcurrentI(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6);
Is2 = LcurrentII(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6);
Is3 = LcurrentIII(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6);
I1 = Is1;
I2 = Is2-Is1;
I3 = Is2;
I4 = Is2-Is3;
I5 = Is3;
I6 = Is2;
end
```

The function *resultLCM* calls the function of the final currents, computes voltage loss for each resistor and then shows all results and given values in the figure of the circuit. The results are displayed in the picture (see fig. 9).

```
function [res] = resultLCM(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6)
```

```
 [I1,I2,I3,I4,I5,I6] =
finalcurrents(R1,R2,R3,R4,R5,R6,V1,V2,V3,V4,V5,V6);
Vr1 = I1*R1;
Vr2 = I2*R2;
Vr3 = I3*R3;
Vr4 = I4*R4;
Vr5 = I5*R5;
Vr6 = I6*R6;
res = imread('circuit.png');
image(res);
axis off
A = sprintf('I(1) = %f A',I1);
text(135,275,num2str(A));
B = sprintf('I(2) = %f A',I2);
text(370,275,num2str(B));
C = sprintf('I(3) = %f A',I3);
text(510,125,num2str(C));
D = sprintf('I(4) = %f A',I4);
text(795,275,num2str(D));
E = sprintf('I(5) = %f A',I5);
text(1150,275,num2str(E));
T = sprintf('I(6) = %f A',I6);
text(350,625,num2str(T));
F = sprintf('%f V',V1);
text(135,500,num2str(F));
R = sprintf('%f V',V2);
text(370,500,num2str(R));
G = sprintf('%f V',V3);
text(370,125,num2str(G));
GH = sprintf('%f V',V4);
text(800,500,num2str(GH));
GL = sprintf('%f V',V5);
text(1158,500,num2str(GL));
GV = sprintf('%f V',V6);
text(650,760,num2str(GV));
H = sprintf('R(1)=%.1f Ohm',R1);
text(135,250,num2str(H));
I = sprintf('R(2)=%.1f Ohm',R2);
text(370,250,num2str(I));
J = sprintf('R(3)=%.1f Ohm',R3);
text(510,100,num2str(J));
K = sprintf('R(4)=%.1f Ohm',R4);
text(795,250,num2str(K));
L = sprintf('R(5)=%.1f Ohm',R5);
text(1150,250,num2str(L));
V = sprintf('R(6)=%.1f Ohm',R6);
text(350,600,num2str(V));
M = sprintf('Vr1=%f V',Vr1);
text(135,300,num2str(M));
N = sprintf('Vr2=%f V',Vr2);
text(370,300,num2str(N));
O = sprintf('Vr3=%f V',Vr3);
text(510,150,num2str(O));
P = sprintf('Vr4=%f V',Vr4);
text(795,300,num2str(P));
Q = sprintf('Vr5=%f V',Vr5);
text(1150,300,num2str(Q));
S = sprintf('Vr6=%f V',Vr6);
text(350,650,num2str(S));end
```

The program is called from the command window using the following command:
[res] = resultLCM(10,20,25,30,15,0,2,0,0,0,3,5);.

The results and given values are illustrated in the following figure (see fig. 9), which appears on the screen after calling the program.

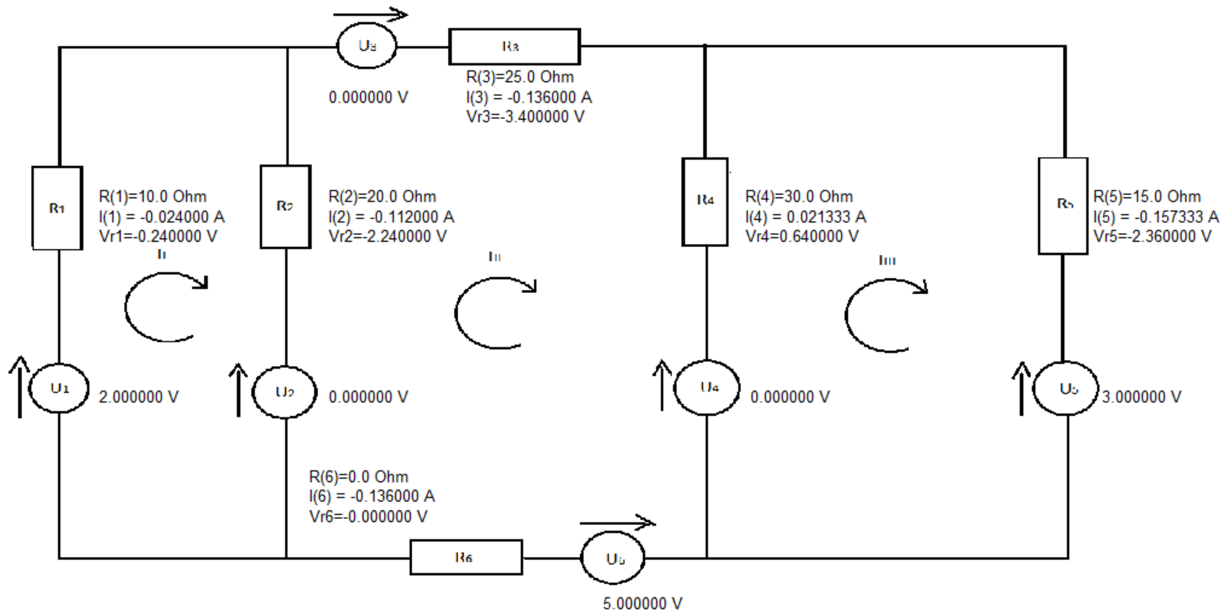


*Figure 9.* Results of the program for the given values.

To improve this program it would be possible to implement all functions in one function or use it for the graphical user interface. It is a quite simple program, and it is sufficient for calculation. It is simple to use – one call in the command window or script is enough to run the program.

## 3.2 Node Voltage Method

First matrices of conductance and currents have to be defined. The matrix of conductance is defined in the function *GmatrixNVM*:

```
function [G] = GmatrixNVM(R1,R2,R3,R4,R5)
```

```
G = [1/R1+1/R2+1/R3,-1/R3;-1/R3,1/R3+1/R4+1/R5];
end
```

The matrix of current is defined in the function *ImatrixNVM*:

```
function [I] = ImatrixNVM(I1,I2,I3,I4,I5)
I = [-I3+I1+I2;+I3+I4+I5];
end
```

The function of the conductance matrix is called in the function *determinant0* and then the determinant of this matrix is computed.

```
function D = determinant0(R1,R2,R3,R4,R5)
Ru0 = GmatrixNVM(R1,R2,R3,R4,R5);
D = det(Ru0);
end
```

The functions of conductance and the current matrix are called in the function *determinant1*. Function changes values of the first column of conductance with values of the current matrix in the for-loop. Then the matrix computes the determinant of the result.

```
function D1 = determinant1(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5)
Ru1 = GmatrixNVM(R1,R2,R3,R4,R5);
I = ImatrixNVM(I1,I2,I3,I4,I5);
for i = 1:2
Ru1(i,1) = I(i,1);
end
D1 = det(Ru1);
end
```

In the following function *determinant2*, the functions of conductance and the current matrix are called. Function changes values of the second column of conductance with values of current matrix in the for-loop. Then the matrix computes the determinant of the result.

```
function D2 = determinant2(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5)
```

```
Ru2 = GmatrixNVM(R1,R2,R3,R4,R5);
I = ImatrixNVM(I1,I2,I3,I4,I5);
for i = 1:2
Ru2(i,2) = I(i,1);
end
D2 = det(Ru2);
end
```

In the following function, the ratio of determinant 1 and 0 is computed.

```
function V10 = voltage10(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5)
D = determinant0(R1,R2,R3,R4,R5);
D1 = determinant1(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5);
V10 = D1/D;
end
```

In the following function, the ratio of determinant 2 and 0 is computed.

```
function V20 = voltage20(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5)
D = determinant0(R1,R2,R3,R4,R5);
D2 = determinant2(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5);
V20 = D2/D;
end
```

In the function *finalvoltages*, voltage losses for every resistor are calculated with respect to the original circuit.

```
function [Vr1,Vr2,Vr3,Vr4,Vr5] =
finalvoltages(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5)
V10 = napeti10(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5);
V20 = napeti20(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5);
Vr1 = V10-I1*R1;
Vr2 = V10-I2*R2;
Vr3 = V20-V10;
Vr4 = V20-I4*R4;
Vr5 = V20-I5*R5;
end
```

Finally, the function *finalvoltages* is called. Currents for each resistor are calculated.

```
function [res] = resultNVM(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5)
 [Vr1,Vr2,Vr3,Vr4,Vr5] = finalvoltages(R1,R2,R3,R4,R5,I1,I2,I3,I4,I5);
Ir1 = Vr1/R1;
Ir2 = Vr2/R2;
Ir3 = Vr3/R3;
Ir4 = Vr4/R4;
Ir5 = Vr5/R5;
res = imread('circuit.png');
image(res);
axis off
A = sprintf('I(1) = %.3f A',I1);
text(210,450,num2str(A));
B = sprintf('I(2) = %.3f A',I2);
text(440,450,num2str(B));
C = sprintf('I(3) = %.3f A',I3);
text(640,110,num2str(C));
B1 = sprintf('I(4) = %.3f A',I4);
text(765,450,num2str(B1));
C1 = sprintf('I(5) = %.3f A',I5);
text(1010,450,num2str(C1));
F = sprintf('Vr1 = %.2f V',Vr1);
text(100,200,num2str(F));
F2 = sprintf('Vr2 = %.2f V',Vr2);
text(320,200,num2str(F2));
G = sprintf('Vr3 = %.2f V',Vr3);
text(500,140,num2str(G));
F1 = sprintf('Vr4 = %.2f V',Vr4);
text(880,180,num2str(F1));
G1 = sprintf('Vr5 = %.2f V',Vr5);
text(1158,240,num2str(G1));
H = sprintf('R(1)=%.1f Ohm',R1);
text(100,160,num2str(H));
I = sprintf('R(2)=%.1f Ohm',R2);
text(320,160,num2str(I));
J = sprintf('R(3)=%.1f Ohm',R3);
text(500,100,num2str(J));
I1 = sprintf('R(4)=%.1f Ohm',R4);
text(880,140,num2str(I1));
J1 = sprintf('R(5)=%.1f Ohm',R5);
text(1158,200,num2str(J1));
```

```
M = sprintf('Ir1=%.3f A',Ir1);
text(100,180,num2str(M));
N = sprintf('Ir2=%.3f A',Ir2);
text(320,180,num2str(N));
O = sprintf('Ir3=%.3f A',Ir3);
text(500,120,num2str(O));
N1 = sprintf('Ir4=%.3f A',Ir4);
text(880,160,num2str(N1));
O1 = sprintf('Ir5=%.3f A',Ir5);
text(1158,220,num2str(O1));
end
```

The program is called in the command window or script using

```
[res] = resultNVM(10,20,25,30,15,0.2,0,0,0.167,0.533);.
```

All results and given values are then displayed in the figure of the original circuit (see fig. 10). This program is used when there are current sources in the circuit. The main disadvantage of this program is that voltage sources are normally used in the electric circuit

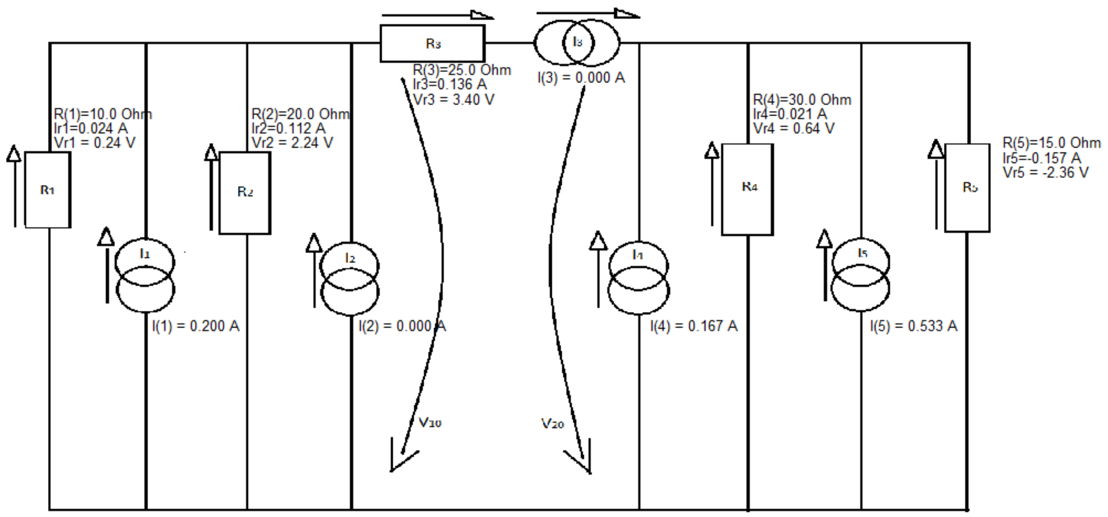


*Figure 10.* Results of the program for the given values.

## 3.3 Kirchhoff's Laws in MATLAB

This program is based on Kirchhoff's laws. It creates an augmented matrix of equations. The inputs of this function are a vector of resistances and a vector of voltages. The vectors have to be given with respect to the universal circuit dependent on the number of loops. If there is a resistance or a voltage in the calculated circuit missing, it has to be replaced in the vector by zero. The function decides which circuit should be calculated on the base of the length of the vector of resistances. An augmented matrix of resistances and voltages is created and saved in the variable *a*. Then a reduced row echelon form of the matrix is made using a function *rref*. This function uses Gaussian elimination to create the matrix with ones on the main diagonal and the last column contains values of the current. The last column of the result is the vector of currents. The function calculates even the voltage losses in each resistor.

```
function [I,Ur] = Kirchhoffslaws(R,V)
switch length(R)
    case 3
         a = [1 1 -1 0;R(1) -R(2) 0 V(1)-V(2);0 R(2) R(3) V(2)-V(3)];
         vysl = rref(a);
         I = vysl(:,4);
         Ur = zeros(1,length(I));
         for i = 1:length(I)
          Ur(i) = R(i)*I(i);
         end
         sprintf('I1 = %f A',I(1));
         sprintf('I2 = %f A',I(2));
         sprintf('I3 = %f A',I(3));
    case 6
          a = [1 1 -1 0 0 0 0;0 0 1 -1 -1 0 0;0 0 0 1 1 -1 0;R(1) -R(2) 0
          0 0 0 V(1)-V(2);...
          0 R(2) R(3) R(4) 0 R(6) V(2)+V(3)-V(4)-V(6);0 0 0 -R(4) R(5) 0
          V(4)-V(5)];
         vysl = rref(a);
         I = vysl(:,7);
         Ur = zeros(1,length(I));
         for i = 1:length(I)
          Ur(i) = R(i)*I(i);
         end
         sprintf('I1 = %f A',I(1));
         sprintf('I2 = %f A',I(2));
         sprintf('I3 = %f A',I(3));
         sprintf('I4 = %f A',I(4));
         sprintf('I5 = %f A',I(5));
         sprintf('I6 = %f A',I(6));
    case 9
          a = [1 1 -1 0 0 0 0 0 0 0;0 0 1 -1 -1 0 0 0 0 0;0 0 0 0 1 -1 -1
          0  0 0;...
          0 0 0 0 0 1 1 -1 0 0;0 0 0 1 0 0 0 1 -1 0;R(1) -R(2) 0 0 0 0 0
          0 0 V(1)-V(2);...
```

```
  0 R(2) R(3) R(4) 0 0 0 0 R(9) V(2)+V(3)-V(4)-V(9);0 0 0 -R(4)
  R(5) R(6) ...
  0 R(8) 0 V(4)+V(5)-V(6)-V(8);0 0 0 0 0 -R(6) R(7) 0 0 V(6)-
  V(7)];
 vysl = rref(a);
 I = vysl(:,10);
 Ur = zeros(1,length(I));
 for i = 1:length(I)
  Ur(i) = R(i)*I(i);
 end
 sprintf('I1 = %f A',I(1));
 sprintf('I2 = %f A',I(2));
 sprintf('I3 = %f A',I(3));
 sprintf('I4 = %f A',I(4));
 sprintf('I5 = %f A',I(5));
 sprintf('I6 = %f A',I(6));
 sprintf('I7 = %f A',I(7));
 sprintf('I8 = %f A',I(8));
 sprintf('I9 = %f A',I(9));
end
end
```

The function is called in the command window:

[I,Ur] = Kirchhoffslaws([10 20 25 30 15 0],[2 0 0 0 3 5]);

A known problem used in the previous chapters will be used to verify whether the function works correctly or not. The results are saved in the workspace and the currents are displayed in the command window.

The result:

I1 = -0.024000 A

I2 = -0.112000 A

I3 = -0.136000 A

I4 = 0.021333 A

I5 = -0.157333 A

I6 = -0.136000 A

There is no doubt whether the results are correct or not. This program will be improved in the chapter about the graphical user interface.

## 3.4 Superposition Method in MATLAB

This program was very complicated. The program for two loops was easy, but it was difficult to make a program for more loops. It would be probably very long and ineffective. The main problem was that some resistances can be equal to 0, so the calculation then has to be different. Also the program for two loops is very long and insufficient for calculations. This program is not reliable enough to be used universally.

This program is programed with respect to the procedure of superposition method. Every voltage loss and every current is calculated as there was only one source. For each source are calculated currents and voltage losses and then they are summed with respect to the given directions.

```
function [I,Vr] = Superposition_method(R,V)
if V(1)==0
            Ic2 = V(2)/((R(1)*R(3)/(R(1)+R(3)))+R(2));
            V22 = Ic2*R(2);
            I21 = (V(2)-V22)/R(1);
            I23 = (V(2)-V22)/R(3);
            V21 = V(2)-V22;
            V23 = V(2)-V22;

            Ic3 = V(3)/((R(2)*R(1)/(R(2)+R(1)))+R(3));
            V33 = Ic3*R(3);
            I32 = (V(3)-V33)/R(2);
            I31 = (V(3)-V33)/R(1);
            V31 = V(3)-V33;
            V32 = V(3)-V33;

            I(1)  = -I21-I31;
            I(2)  = +Ic2-I32;
            I(3)  = -I23+Ic3;
            Vr(1)  = -V21-V31;
            Vr(2)  = V22-V32;
            Vr(3)  = -V23+V33;

 elseif V(2)==0

            Ic1 = V(1)/((R(2)*R(3)/(R(2)+R(3)))+R(1));
            V11 = Ic1*R(1);
            I12 = (V(1)-V11)/R(2);
            I13 = (V(1)-V11)/R(3);
            V12 = V(1)-V11;
            V13 = V(1)-V11;

            Ic3 = V(3)/((R(2)*R(1)/(R(2)+R(1)))+R(3));
            V33 = Ic3*R(3);
            I32 = (V(3)-V33)/R(2);
            I31 = (V(3)-V33)/R(1);
            V32 = V(3)-V33;
            V31 = V(3)-V33;

            I(1)  = Ic1-I31;
            I(2)  = -I12-I32;
            I(3)  = -I13+Ic3;
            Vr(1)  = V11-V31;
            Vr(2)  = -V12-V32;
            Vr(3)  = -V13+V33;
```

```
elseif V(3)==0

        Ic1 = V(1)/((R(2)*R(3)/(R(2)+R(3)))+R(1));
        V11 = Ic1*R(1);
        I12 = (V(1)-V11)/R(2);
        I13 = (V(1)-V11)/R(3);
        V12 = V(1)-V11;
        V13 = V(1)-V11;

        Ic2 = V(2)/((R(3)*R(1)/(R(3)+R(1)))+R(2));
        V22 = Ic2*R(2);
        I23 = (V(3)-V33)/R(3);
        I21 = (V(3)-V33)/R(1);
        V23 = V(3)-V33;
        V21 = V(3)-V33;

        I(1) = Ic1-I21;
        I(2) = -I12+Ic2;
        I(3) = -I13-I23;
        Vr(1) = V11-V21;
        Vr(2) = -V12+V22;
        Vr(3) = -V13-V23;

elseif V(1)==0 && V(2)==0

        I(3) = V(3)/((R(2)*R(1)/(R(2)+R(1)))+R(3));
        Vr(3) = R(3)*I(3);
        I(1) = (V(3)-Vr(3))/R(1);
        Vr(1) = R(1)*I(1);
        I(2) = (V(3)-Vr(3))/R(2);
        Vr(2) = R(2)*I(2);

elseif V(1)==0 && V(3)==0

        I(2) = V(2)/((R(3)*R(1)/(R(3)+R(1)))+R(2));
        Vr(2) = R(2)*I(2);
        I(1) = (V(2)-Vr(2))/R(1);
        Vr(1) = R(1)*I(1);
        I(3) = (V(2)-Vr(2))/R(3);
        Vr(3) = R(3)*I(3);

elseif V(2)==0 && V(3)==0

        I(1) = V(1)/((R(3)*R(2)/(R(3)+R(2)))+R(1));
        Vr(1) = R(1)*I(1);
        I(2) = (V(1)-Vr(1))/R(2);
        Vr(2) = R(2)*I(2);
        I(3) = (V(1)-Vr(1))/R(3);
        Vr(3) = R(3)*I(3);

else

        Ic1 = V(1)/((R(2)*R(3)/(R(2)+R(3)))+R(1));
        V11 = Ic1*R(1);
        I12 = (V(1)-V11)/R(2);
        I13 = (V(1)-V11)/R(3);
        V12 = V(1)-V11;
        V13 = V(1)-V11;

        Ic2 = V(2)/((R(1)*R(3)/(R(1)+R(3)))+R(2));
        V22 = Ic2*R(2);
        I21 = (V(2)-V22)/R(1);
        I23 = (V(2)-V22)/R(3);
        V21 = V(2)-V22;
        V23 = V(2)-V22;

        Ic3 = V(3)/((R(2)*R(1)/(R(2)+R(1)))+R(3));
        V33 = Ic3*R(3);
        I32 = (V(3)-V33)/R(2);
        I31 = (V(3)-V33)/R(1);
```

```
                V32 = V(3)-V33;
                V31 = V(3)-V33;

                I(1) = Ic1-I21-I31;
                I(2) = -I12+Ic2-I32;
                I(3) = -I13-I23+Ic3;
                Vr(1) = V11-V21-V31;
                Vr(2) = -V12+V22-V32;
                Vr(3) = -V13-V23+V33;
    end
    end
```

This program calculates the circuit separately for every possible combination of voltage sources. It calculates the circuit differently, when one or two of the components of the calculated circuit are missing. It was hard to create program for every possible combination. The program is done for one, two, or three sources and three resistors. If there is only one source in calculated circuit, it is easy to use simplification method for DC circuit analysis, but this program calculates even this possibility. The program cannot calculate the circuit, when one or more resistors are missing, the results are not correct. In this case other programs are recommended to use. The program uses no function from MATLAB function. It uses only simple mathematical operation such as sum, subtraction, division and multiplication. It is called in the command window using:

```
[I1,Vr1] = Superposition_method([100 200 300],[5 0 3])
```

The results are then:

```
I1 =

    0.0173   -0.0164   -0.0009

Vr1 =

    1.7273   -3.2727   -0.2727
```

To be sure, whether the results are correct or not, program for Kirchhoff's laws can be used. It is called from the command window using:

```
[I2,Vr2] = Kirchhoffslaws([100 200 300],[5 0 3])
```

The results are then:

```
I2 =

    0.0173
   -0.0164
    0.0009

Vr2 =

    1.7274   -3.2727    0.2727
```

So the results are correct.

# 4 Graphical User Interface

There are three types of user interfaces: command window, text user interface and graphical user interface. In the third chapter was the first user interface used. All those programs were called from the command window. According to MathWorks (n.d.):

GUIs (also known as graphical user interfaces or UIs) provide point-and-click control of software applications, eliminating the need to learn a language or type commands in order to run the application.
MATLAB® apps are self-contained MATLAB programs with GUI front ends that automate a task or calculation. The GUI typically contains controls such as menus, toolbars, buttons, and sliders. Many MATLAB products, such as Curve Fitting Toolbox™, Signal Processing Toolbox™, and Control System Toolbox™ include apps with custom user interfaces. You can also create your own custom apps, including their corresponding UIs, for others to use.

According to Škutková (2016), the graphical user interface is a set of practices that affect the behaviour of machines, devices, computer programs, or complex systems. Škutková (2016) also points out that graphical user interface has to have specific properties: It should be simple and intuitive. It is intended for ordinary users, not just for professionals. It helps to achieve a more effective goal. It makes easier working with the system, but does not ensure the proper functioning. It helps maintain integrity and data security.

The graphical user interface uses rastro-oriented imaging devices. It should display a faithful image of the printed output. It uses graphical controls (icons, buttons, windows, etc.). The graphical user interface should be user-friendly. It allows direct manipulation of on-screen objects. It allows control in a uniform manner.

Consistent terminology and control have to be maintained. The user interface has to be customized with a wide range of users. Feedback to the user should be provided. The graphical user interface should navigate the user. Any control should be predicable.

## 4.1 Kirchhoff's laws in MATLAB GUI (Graphical User Interface)

The graphical user interface in MATLAB is made using a command *guide* in the command window. A new figure and MATLAB code (m-file) is created then. A design of the graphical user interface can be made using *guide.*

Many editable text boxes have to be used, because the user needs to set the input resistances and voltages. Also the user should see the circuit which they calculate. There also has to be a button which tells the program to calculate currents in the circuit. The result should appear on the screen. The design should be as simple as possible. First design of the program has to be made. In the following figure a layout of all components can be seen (see fig. 11).

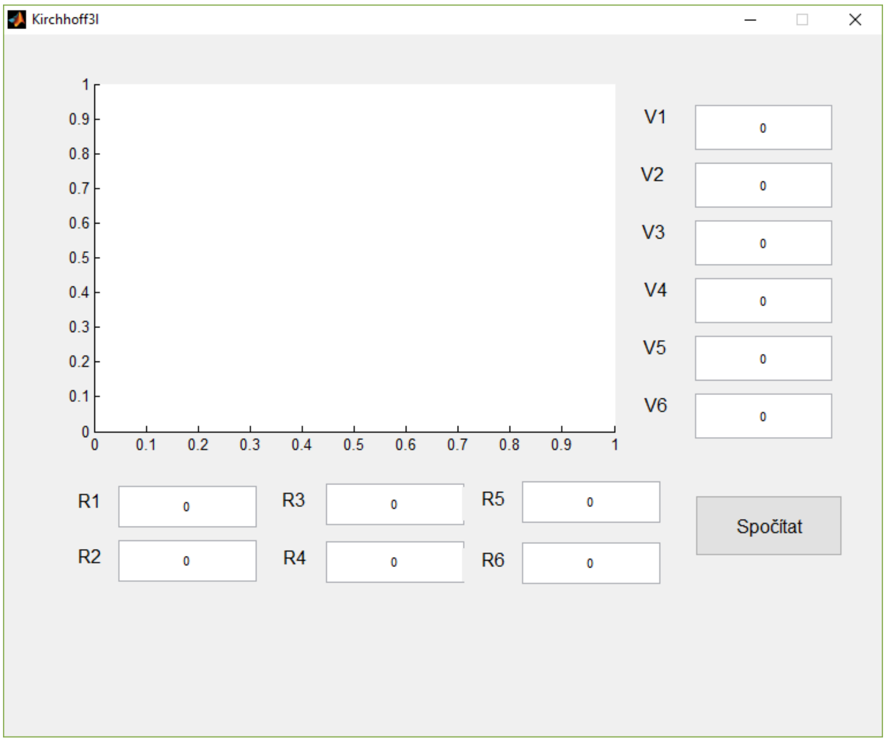


*Figure 11*. Layout of all components in the figure.

The first step is to improve the opening function of the program, because it is needed for the program to appear the picture of the circuit in the window. It will be implemented using the function *imshow* as following.

```
axes(handles.axes1)
```

```
G = imread('circuit31.bmp');
imshow(G)
```

The next step is to implement the program into the m-file. The program is controlled especially with two functions – *get* and *set*. The function *get* takes the content of any object in the graphical design. First the content of editable text boxes has to be gotten.

```
R1 = str2double(get(handles.edit1,'String'));
R2 = str2double(get(handles.edit2,'String'));
R3 = str2double(get(handles.edit3,'String'));
R4 = str2double(get(handles.edit4,'String'));
R5 = str2double(get(handles.edit5,'String'));
R6 = str2double(get(handles.edit6,'String'));
V1 = str2double(get(handles.edit8,'String'));
V2 = str2double(get(handles.edit9,'String'));
V3 = str2double(get(handles.edit10,'String'));
V4 = str2double(get(handles.edit11,'String'));
V5 = str2double(get(handles.edit12,'String'));
V6 = str2double(get(handles.edit13,'String'));
```

The function *set* sets any object specific properties (changes the content, sets visibility, etc.). After the calculation of values of the currents the values have to be shown as the results in the static text boxes. The values will be implemented using the function *get* as following.

```
text1 = sprintf('I1 = %f A',I1);
text2 = sprintf('I2 = %f A',I2);
text3 = sprintf('I3 = %f A',I3);
text4 = sprintf('I4 = %f A',I4);
text5 = sprintf('I5 = %f A',I5);
text6 = sprintf('I6 = %f A',I6);
set(handles.text13,'String',text1);
set(handles.text14,'String',text2);
set(handles.text15,'String',text3);
set(handles.text16,'String',text4);
set(handles.text17,'String',text5);
set(handles.text18,'String',text6);
```

In this case, previously created program for Kirchhoff's laws is used. In the following figure the program ready for use can be seen (see fig. 12).

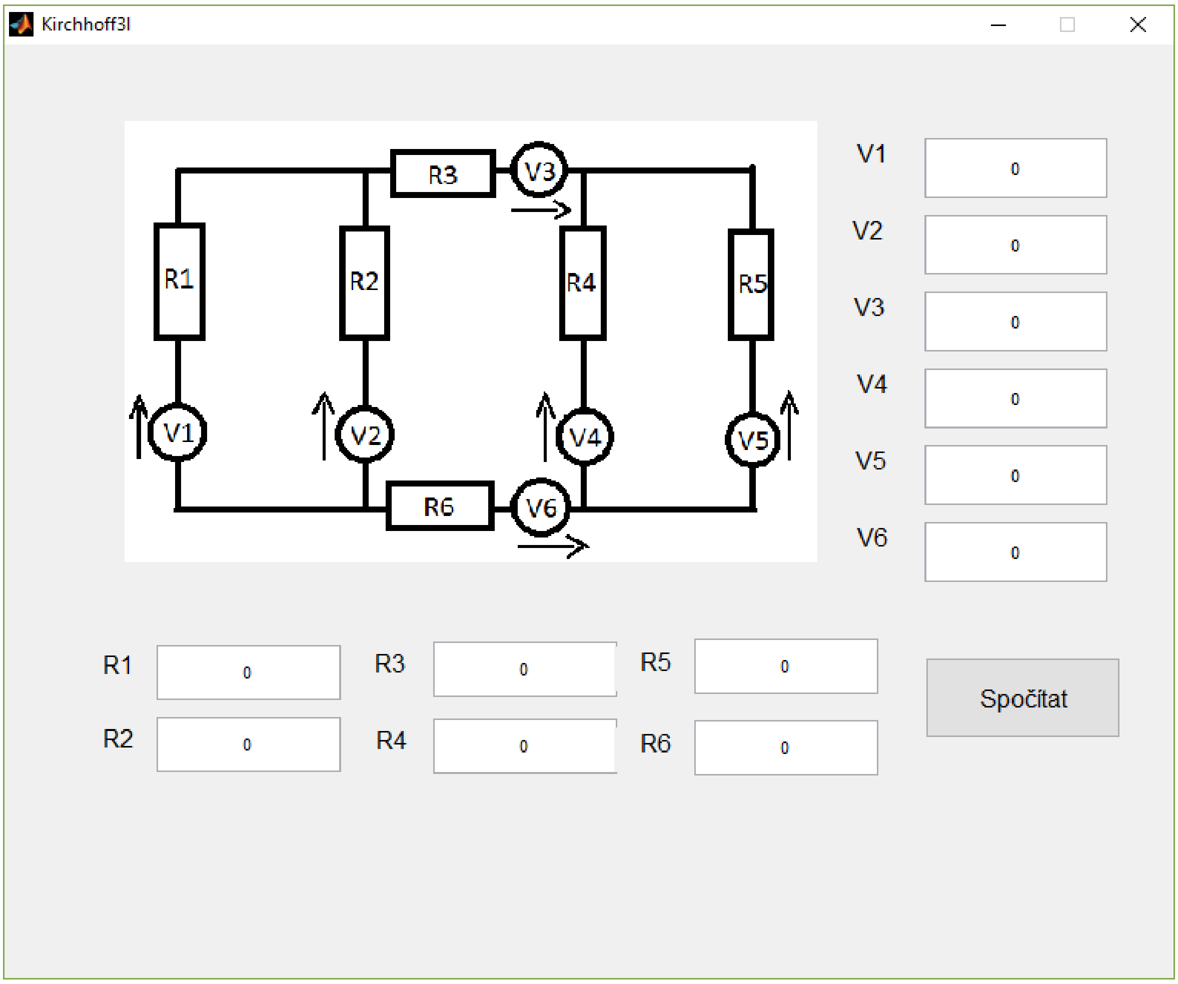


*Figure 12*. Program ready for use.

The GUI has to be user friendly, so it is obligatory to have contextual help implemented. Inputs need to be treated. The user should be able to fix it and should be warned by the program if the input is in an incorrect format. The program has to communicate with the user. The program after incorrect inputs can be seen in the following figure (see fig. 13). The correction of the inputs is possible.

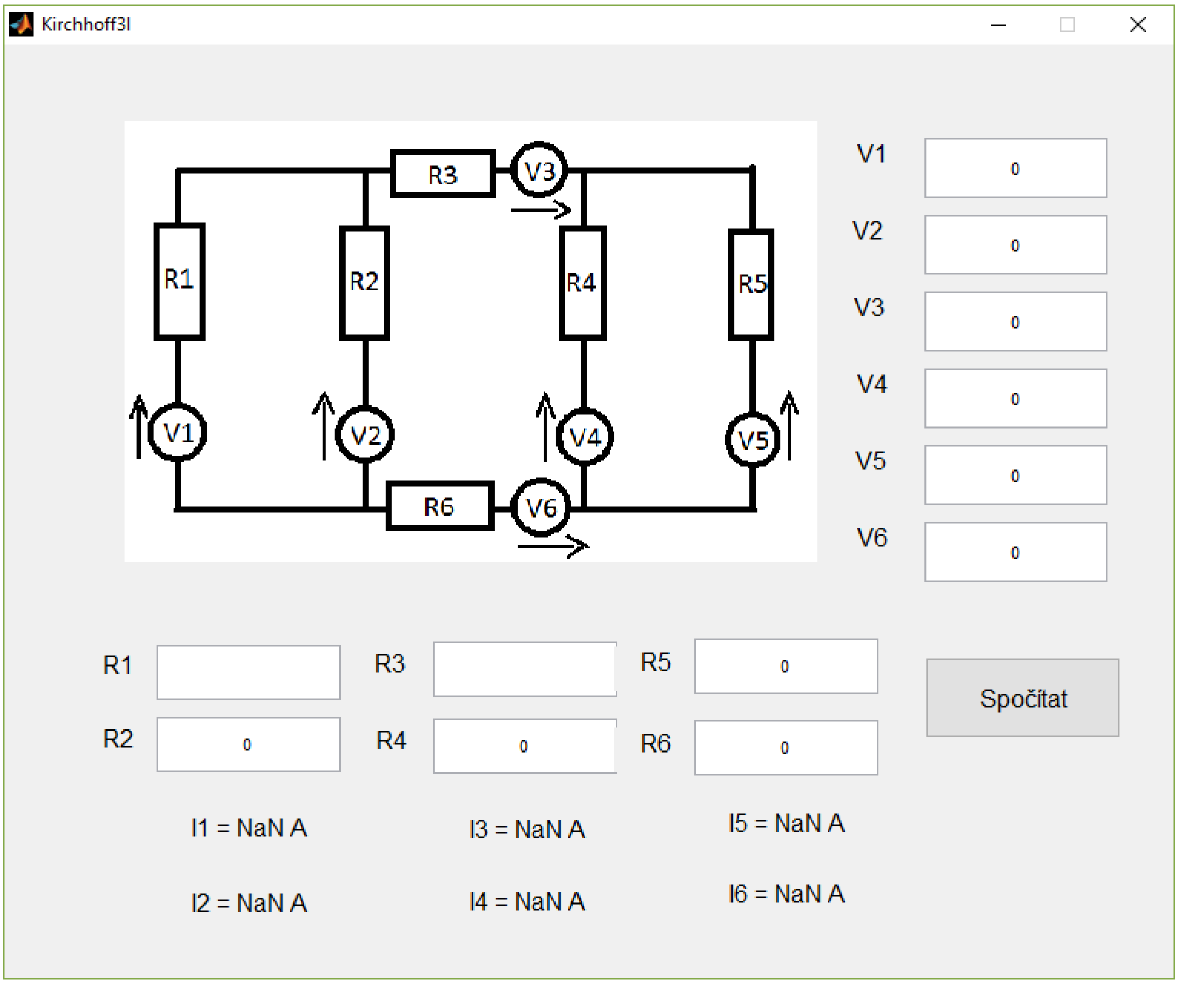


*Figure 13*. Program with wrong inputs.

When there is no resistor or no voltage on the specific position in the calculated circuit it has to be replaced by zero. Otherwise the program cannot calculate the right results (see fig. 13). To control whether the results are right or not the example from the previous chapters will be used. The program after given correct inputs can be seen in the following figure (see fig. 14).

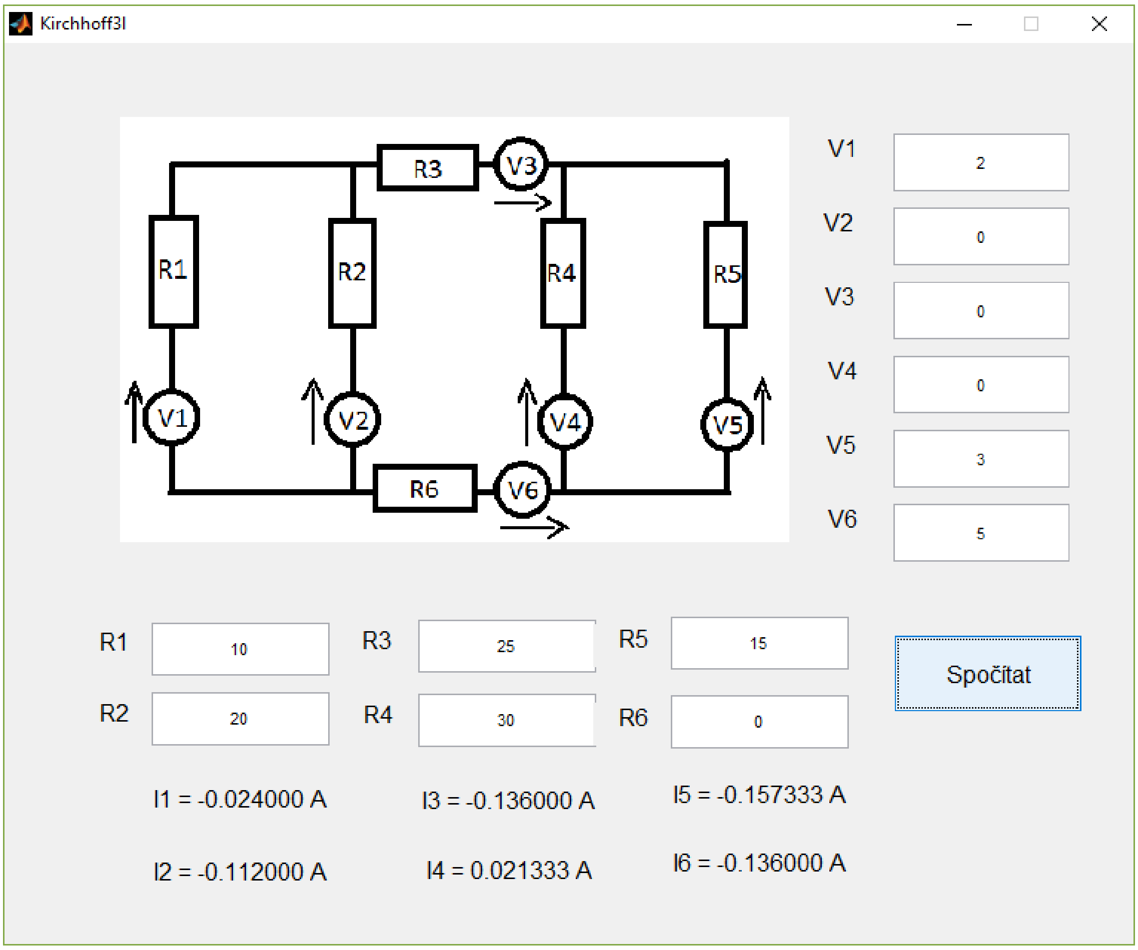


*Figure 14.* Results of the graphical user interface.

This program is easy to use. It has right results in the case of correctly given inputs. It is well-arranged so the user can easily orientate in the program. *Tooltip string* is used to inform the user which units should be given in each text box. When the user uses their mouse and stop on the position, where editable text box is, it will show the instruction, what the input is and in which unit it should be given. Inputs are possible correct, if they are given in the wrong way. The program is run using one button in the MATLAB window.

To improve the program it would be possible to calculate voltage losses in each resistor and show them as results below the currents. Next possibility could be better more colourful design. This design is very simple.

The main advantage of the program is that it is very simple and it is easy to use. The main disadvantage of this program is that the design is too simple.

## 4.2 Other possibilities

In this subchapter, other possibilities of a program with the graphical user interface will be discussed. The programs were written with respect to the particular methods. These programs can be improved. There could be a universal program with all those methods combined together in order to compute every circuit with any combination of voltage and current sources.

These programs can also be modified for more circuit components such as capacitors or inductors. It would need to be calculated with the complex numbers and it would be AC circuit analysis, not DC.  The program would be then more universal.

Considering the possibilities of other programming languages, MIT App Inventor has to be mentioned. According to MIT App Inventor (n.d.):

> MIT App Inventor is an innovative beginner's introduction to programming and app creation that transforms the complex language of text-based coding into visual, drag-and-drop building blocks. The simple graphical interface grants even an inexperienced novice the ability to create a basic, fully functional app within an hour or less.

Applications for smartphones are created in this programming environment and programming language. This programming language is not text-based, but it is in the form of drag-and-drop boxes. This form is easy to understand and program in it, but it lasts longer than normal text-based programming. This application could have been very useful, because the user does not need to have MATLAB installed in his computer. The disadvantage of this programming environment is that the programmer has very restricted possibilities. The main advantage is that almost everyone has a smartphone and uses these kinds of applications. These applications are in the format of *.apk* so it is suitable especially for Android.

# 5 Conclusion

In this Bachelor's thesis, the different methods of DC analysis were discussed. Four of these methods were transferred into the MATLAB programming language. In the conclusion, the methods should be compared. Universal methods can be used for every DC linear circuit, but special methods are only for the specific examples. This is the greatest disadvantage of special methods, but for example, the simplification method is simpler to calculate than the other methods if a circuit with one source is analysed. The simplification method and Kirchhoff's laws are used with combination of other methods because they are part of them.

Some of these methods are complicated to calculate by hand, especially when the calculated circuit is very complicated, with many loops and nodes. Then it is easier to use the program at least for the verification of the solution. Everyone sometimes forgets minus somewhere and then the solution is completely wrong. For this reason, the programs for the verification are very useful.
To compare the programs of the node voltage method and the loop current method, I think the loop current method is more universal than the node voltage method, because in practice, there more voltage sources and resistors than current sources are used and conductance is not a real component here. Both loop current method and node voltage method are simple and well-working programs. Both of them can be improved using only one function variable with respect to the number of loops or nodes as it was programed for Kirchhoff's laws. The program for the superposition method is ineffective and insufficient for calculation. The program for Kirchhoff's laws is the most universal, so it was also modified for the graphical user interface. The program itself is very short and simple. The graphical user interface is better for the students who cannot effectively work with the command window in MATLAB. The program created in the graphical user interface is very simple and effective. In the last chapter other possibilities for programs were discussed. All programs were tested using a known example.

# List of References


Herrick, Robert J. (2003). *DC/AC Circuits and Electronics: Principles & Applications.* Clifton Park: Thomson/Delmar Learning

Chen, Wai-Kai. (2005). *The Electrical Engineering Handbook.* Boston: Elsevier Academic Press.

Doňar, B., & Zaplatílek, K. (2004). *MATLAB – tvorba uživatelských aplikací*. Praha: BEN – technická literatura.

Wang, M. (2010). *Understandable electric circuits.* London: Institution of Engineering and Technology.

Myslík, J. (1979). *Řešené příklady z analýzy elektrických obvodů.* Plzeň: Vysoká škola strojní a elektrotechnická.

Škutková, H. (2016). *Uživatelské rozhraní aplikací v biomedicíně*. Brno: Vysoké Učení Technické v Brně.

MATLAB GUI [Online]. (n.d.) Retrieved May 15, 2017 from https://www.mathworks.com/discovery/matlab-gui.html?s_tid=srchtitle

What is MIT App Inventor [Online]. (n.d.) Retrieved May 15, 2017 from http://appinventor.mit.edu/explore/about-us.html

Farrell, C. (1976). 150 years from Ohm's Law. *New Scientist, 71*(1020), 704-705.

Maloberti, F., & Davies, A. C. (Eds.). (2016). *A Short History of Circuits and Systems*. Gistrup: River Publishers.

Attia, J. O. (1999). *Electronics and Circuit Analysis using MATLAB*. New York: CRC Press.

Asad, Y. et al. (2014). *Electric Circuit Analysis in MATLAB and Simulink* (1st ed.). Indianapolis: American Society for Engineering Education.

## List of Figures

## List of Equations

$I_1 + I_2 - I_3 = 0$ (1)

$I_3 + I_4 - I_5 = 0$ (2)

$R_1 \bullet I_1 - R_2 \bullet I_2 = V_1$ (3)

$R_2 \bullet I_2 - R_3 \bullet I_3 - R_4 \bullet I_4 = - V_2$ (4)

$R_4 \bullet I_4 - R_5 \bullet I_5 = - V_3$ (5)

$$\begin{bmatrix} R1+R2 & -R2 & 0 \\ -R2 & R2+R3+R4 & -R4 \\ 0 & -R4 & R4+R5 \end{bmatrix} \cdot \begin{bmatrix} I_I \\ I_{II} \\ I_{III} \end{bmatrix} = \begin{bmatrix} V_1 \\ -V_2 \\ -V_3 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} V_1 & -R2 & 0 \\ -V_2 & R2+R3+R4 & -R4 \\ -V_3 & -R4 & R4+R5 \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} R1+R2 & V_1 & 0 \\ -R2 & -V_2 & -R4 \\ 0 & -V_3 & R4+R5 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} R1+R2 & -R2 & V_1 \\ -R2 & R2+R3+R4 & -V_2 \\ 0 & -R4 & -V_3 \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} \frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3} & \frac{-1}{R3} \\ \frac{-1}{R3} & \frac{1}{R3}+\frac{1}{R4}+\frac{1}{R5} \end{bmatrix} \cdot \begin{bmatrix} V10 \\ V20 \end{bmatrix} = \begin{bmatrix} -I1 \\ -I2-I3 \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} \frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3} & \frac{-1}{R3} \\ \frac{-1}{R3} & \frac{1}{R3}+\frac{1}{R4}+\frac{1}{R5} \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} -I1 & \frac{-1}{R3} \\ -I2-I3 & \frac{1}{R3}+\frac{1}{R4}+\frac{1}{R5} \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} \frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3} & -I1 \\ \frac{-1}{R3} & -I2-I3 \end{bmatrix} \tag{13}$$

$$R_1 = \frac{R_A R_C}{R_A + R_B + R_C} \tag{14}$$

$$R_2 = \frac{R_A R_B}{R_A + R_B + R_C} \tag{15}$$

$$R_3 = \frac{R_B R_C}{R_A + R_B + R_C} \tag{16}$$

$$R_A = R_1 + R_2 + \frac{R_1 R_2}{R_3} \tag{17}$$

$$R_B = R_2 + R_3 + \frac{R_2 R_3}{R_1} \tag{18}$$

$$R_C = R_1 + R_3 + \frac{R_1 R_3}{R_2} \tag{19}$$